DIGITAL SPEECH SYNTHESIZER

By

AKALANK H. KILLEDAR

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

DIGITAL SPEECH SYNTHESIZER

A Thesis Submitted
In Partial Fulfilment of the Requirements
for the Degree of
MASTER OF TECHNOLOGY

By
AKALANK H. KILLEDAR

to the

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

Digital Speech Synthesizer

A Thesis Submitted
In Partial Fulfilment of the Requirements
for the Degree of

MASTER OF TECHNOLOGY

by

Amalank H. Killedar

to the

Department of Electrical Engineering
Indian Institute of Technology
KANPUR.

CERTIFICATE

Certified that this thesis entitled 'Digital Speech Synthesizer' has been carried out under my supervision by Mr. Akalank H. Killedar and has not been submitted elsewhere for a degree.

R Biswas
(R.N. Biswas)
Assistant Professor,
Department of Electrical Engineering,
Indian Institute of Technology,
KANPUR.

July, 1975.

ACKNOWLEDGEMENT

I wish to express my deep sense of gratitude to Dr. R.N. Biswas for suggesting this interesting project and for many long hours of helpful discussions on the same.

I am also grateful to Dr. N. Ramasubramanian for giving vital references and clearing the ideas about the problem. My special thanks are due to the staff and faculty of E.E. department who directly, or indirectly helped in completing the project.

I would also like to thank Mr. C.M. Abraham for typing out this report with patience and care at a short notice.

Kanpur,
July, 1975.

Akalank H. Killedar

CONTENTS

ABSTRACT

A digital filter system for use in speech synthesis has been designed and fabricated using the timeshared arithmetic. The synthesizer consists of 2 paths: the upper path for voiced speech having six 2-pole filters and one 2-zero filter in cascade and the lower path for unvoiced speech having two 2-pole filters and one 2-zero filter in cascade. One output sample requires each of the ten filters containing two words. The word-length has been chosen to be 24 bits from the consideration of truncation errors. One complete iteration thus requires 480 (20 x 24) clocks. The 16-bits coefficients for multiplicand can be provided either from a real time on-line computer or from switches. 2's complement integer arithmetic has been used throughout. Sufficient flexibility has been maintained in the hardware so that it can as well be used as a bank of digital filters.

Chapter 1

INTRODUCTION

Over the past fifteen years many attempts have been made to devise schemes for the synthesis of natural sounding, intelligible speech from a linguistic description of an utterance. Speech synthesis is an important means of testing theories of human speech production. Speech synthesizers are valuable tools for better understanding of the human speech production process and for perception studies . Among other applications in speech research are investigation of computer voice response, reading machines for the blind,vocoders, speech production through typewriter input and computer assisted instructions etc.

1.1 MECHANISM OF SPEECH PRODUCTION : -

Speech synthesizer is a simulation of vocal tract proper which is an acoustical tube with non uniform cross-sectional area, terminated at one end by lips and the other by the vocal cord constriction at the top of the trachea (wind pipe). The vocal tube is deformed in cross-section by the movement of articulators; namely, the lips, tongue and velum. Nasal tract is an auxiliary path for sound transmission. It begins at the velum and terminates at nostrils.

The velum controls the acoustic coupling. Air is forced from lungs through trachea and vocal cord and vocal tract. Vibratory action (resembling pulse generator output) of vocal cord together with various constrictions in the tract produce sounds of speech.

1.2 SPEECH SYNTHESIS STRATEGIES : -

The different synthesis strategies used are formant synthesis, synthesis from printed text and articulatory synthesis. The first one is implemented in this project. Formant synthesizer can be simulated in 3 ways - computer simulation, terminal analog synthesizer, digital synthesizer. Real time response in the first case is very difficult. Most of the general purpose computers take 2 to 20 times real time required for speech synthesis [2]. The digital synthesizer has many advantages over its analog counterpart, e.g. precise control over centre frequencies and bandwidths of resonators, reliability and stability of hardware, light weight, small size and low power consumption.

1.3 HARDWARE USED AS DIGITAL FILTER BANK : -

As will be shown in the later chapters, the synthesizer mainly consists of various 2 pole,2 zero digital filters cascaded in a particular fashion. Hence the hardware is essentially a set of digital filters and can be used as such for other applications as well.

The hardware fabricated resembles the one described by L.B. Jackson for Rockland System Corporation [2] except for the path for voiced fricatives (explained in next chapter) and 320 bit coefficient memory. The control circuitry is also changed for the purpose described in the above paragraph.

Chapter 2 is concerned with characteristic components of speech, digital transfer function for phoneme types, spectral compensation and general synthesizer schematic. Chapter 3 deals with the problems of different hardware realizations, errors due to finite wordlength; input-output specifications and the time sharing concept used in realization. In Chapter 4, design aspects of various hardware blocks are elucidated, while Chapter 5 shows the results of hardware testing. Chapter 6 concludes the thesis with remarks about required peripherals, suggested modifications and so on. The appendices include the Z-transforms, phoneme characteristic formants and coefficients, 2's complement multiplication and printed circuit connection diagram.

Chapter 2

SPEECH CHARACTERIZATION AND GENERAL SCHEMATIC

In this chapter we deal with the different characteristics of speech which are very important in synthesis problems. Description of various phonemes, their formants and formant motion with changing phonemes, pitch etc. is given. Digital T.F. for different types of phonemes i.e., vowels; fricatives - voiced, unvoiced, nasals sounds etc. are given as well as that for spectral compensation. At the end schematic diagram of the formant synthesizer is explained.

2.1 SPEECH CHARACTERIZATION : -

Any language consists of a finite number of distinguishable, mutually exclusive sounds which form the basic linguistic elements called phonemes such that if one replaces another in utterance, the meaning is changed. In short, phonemes can be thought of as uniquely related to the articulatory gestures of a language. The phonemo classificatior of sounds of English speech in general American (GA) dialect and their characterization is shown below :

2.1.1 Phoneme Characteristics [1] : Phonemes can be generally classified into (a) vowels (b) consonants and (c) semivowels. The consonants themselves are classified as stops, nasals etc. Their characterization is given below.

(1) Vowels (a,e,i,o,u) :- These are normally produced by vocal cord excitation of the tract, the tract having relatively stable configuration, usually negligible nasal coupling and radiation only from mouth. It gives fairly steady state energy bands (formants) at characteristic frequencies.

(2) Consonants :- These are characterized by greater vocal tract constrictions and may be excited and/or radiated differently.

(a) Fricatives (s,sh,z,zh,v,f):- These have incoherent noise excitation of the vocal tract.- Greatest point of constriction is between tongue and lips. Radiation is from the mouth. For voiced fricatives, vocal cord also vibrates.

(b) Stops (p,t,k,b,d,g) :- These are characterized by short period of silence as the lips or tongue block off the airflow, followed by a short burst as the pent up air is released.

(c) Nasals (m,n,ng) :- Here the velum is opened wide for nasal coupling. At the end, front of vocal tract completely closes.

(3) Glides and Semivowels (y,w,l,r) :- Their utterance resemble that of vowels. They have voiced excitation, radiation from mouth and produce flowing clear sounds.

2.1.2 Formants and Formant Motion [4] :- Spectrograms of a continuous speech show three different resonances, and distinct too. Their centre frequencies and bandwidths vary depending upon the phonemes, context, preceeding and following phonemes. There are higher resonances too, which have relatively fixed centre frequencies (usually spaced equidistant apart on frequency axis) and bandwidths. These resonances are termed formants. The table in appendix B shows first 3 formants for various phonemes and computed coefficients according to relations shown later in this chapter.

Motion of formants very significantly contributes to the intelligibility of speech. Smooth, continuous formant transitions, as observed in spectrograms of real speech, are described by a critically damped second degree differential equation. Formants, ingeneral, are in motion towards target values of the phonemes tobe generated. Each of the formant moves with its own time constant and there is provision for delay in time of initiation of the formant motion.

2.1.3 Pitch :- Voice pitch is the period of vibration of the vocal cord which depends partly upon excitation. The voice excitation itself is quasi periodic. The waveform varies in period, amplitude as well as in shape [1]. Triangular waveform is a common one used in synthesis of sounds.

2.2 DIGITAL TRANSFER FUNCTIONS FOR SPEECH CHARACTERISTICS :-

Any digital transfer function consists of 2 basic building blocks having transfer functions as indicated below :

$$\text{TWO-POLE RESONATOR : } H_p(z^{-1}) = \frac{1-2\ e^{-aT}\cos bT + e^{-2aT}}{1-2\ e^{-aT}\cos bT\ z^{-1} + e^{-2aT}z^{-2}} \qquad (2.1)$$

$$\text{TWO-ZERO ANTIRESONATOR : } H_z(z^{-1}) = \frac{1-2\ e^{-aT}\cos bT\ z^{-1} + e^{-2aT}z^{-2}}{1-2\ e^{-aT}\cos bT + e^{-2aT}} \qquad (2.2)$$

Note that the constant terms are for unity transmission at zero frequency . The variables are :

a = radian bandwidth of pole/zero.

b = radian centre frequency of pole/zero.

T = sampling period.

z^{-1} = e^{-sT} = unit delay operator.

This section deals with the digital transfer functions required for pulse shaping, spectral compensation, formants viz. vowels, fricatives and nasals. The pole-zero requirements for these various functions are listed below:

(1) Formants of non-nasal sounds :- These sounds have 3 different formant frequencies and corresponding bandwidths and hence require 3 2-pole resonantors.

(2) Spectral compensation :- The effects of radiation of sound from mouth or nose into air and glottal rexcitation pulse shape are accounted for by the fixed spectral compensation network consisting of 2 2-pole resonators.

(3) Formants of Nasal sounds :- For nasal sounds, in addition to the formant resonators, a 2-pole resonator , and a 2-zero antiresonator are required. During non-nawal sounds, these pole-zero pairs cancel out. In this case, the resonator may be used for higher pole correction.

The unity gain at zero frequency is to account for the unity transmission of the vocal tract at zero frequency. All the transfer functions are derived from the matched Z-transform [3, Appendix A].

2.3 INPUT REQUIREMENTS : -

Depending upon the phonemes tobe generated, either of the following inputs are required for the digital filters.

2.3.1 The Pitch Pulse Generator :- It provides excitation for voiced speech, which includes vowels as well as both voiced and unvoiced components of voiced fricatives. Normally the pitch varies its duration between 5 millisec. to 20 millisec. The actual hardware may vary depending upon the type of pulses wanted (analog or digital - triangular or square or trapezoidal etc.). The gain of the voiced path determines the amplitude of the pulse to represent intensity of sound.

2.3.2 The Frication Generator :- It is essentially a white noise generator approximated in the digital circuitry by a pseudo random number generator. One of the algorithms to generate these numbers is a 16 bit maximal length shift register sequence whose current input bit is defined by

$$x_n = x_{n-1} \oplus x_{n-12} \oplus x_{n-14} \oplus x_{n-15}, \quad n = 0,1,2 \;.. \quad (2.3)$$

The frication generator is a source of aspiration, whispered speech and frication noise for fricatives and voiceless stop consonants.

2.4 THE SYNTHESIZER OUTPUT :-

The outputs of the different filters used to produce vowels, consonants and semivowels are added in the accumulator to give digital output which can be returned to the computer for further processing and also fed to the D/A converter for analog speech output. The different paths are described below,

2.4.1 Path for Voiced Speech : - Voiced speech, generated by vocal cord vibration, requires pulse input to a combination of the following seven blocks.

a) 2 resonators to account for the pulse shaping and spectral compensation network, usually having fixed centre frequencies and bandwidths.

b) 3 resonators for the first 3 formants of voiced sounds, usually of varying centre frequencies and bandwidths to account for the time varying shape of the vocal tract on speech spectrum, and;

c) One resonator and one antiresonator to account for the nasal poles and zeros. During the non-nasal sounds this resonator can be used for higher pole correction. Higher pole correction is especially easy for sampled data system since a pole at any frequency represents an infinite number of poles with a spacing equal to the sampling frequency [Appendix A]

2.4.2 Path for Fricative Sounds :- A noise amplitude controls the noise input to the fricative network consisting of a resonator and an antiresonator. Another resonator is used for spectral compensation, Here generally the bandwidths are fixed, only the centre frequencies vary.

2.4.3 Path for Voiced Fricatives :- This path is used to produce voiced fricatives [6,7]. A threshold level is subtracted from the output of the second resonator of 2.4.1. (analog of volume velocity from the glottis through the point of constriction of the vocal tract) and only the positive part of the result is transmitted. The operation models the physical fact that turbulance is produced only when the volume velocity of the airflow in vocal tract exceeds a threshold value. This output modulates the noise to produce pitch-synchronous noise excitation.

Chapter 3

HARDWARE SCHEMATIC AND TIMESHARING

This chapter deals with the comparison of parallel an and serial realizations of the synthesizer, input - output specifications and timesharing concept used to implement all the digital filters with a single arithmetic unit.

3.1 PARALLEL VERSUS SERIAL REALIZATION : -

The different paths described in the previous chapter can either be implemented by using parallel realization or by using serial realization. This results from the fact that any transfer function can be expressed either

as a sum of the partial fractions form
or as a product of second order transfer functions.

These two possibilities correspond respectively to parallel and cascade combinations of second-order filters. The former realization will have independent gain controls for each resonator, whereas the later one will have only one (or two for two sources) gain control. This reduces the complexity of rules for synthesis (i.e., the software is simplified). Vowel spectra from the parallel synthesizer produces extraneous zeros between resonances, which may cause incorrect overall output. Hence they have no higher pole correction networks but the zeros are used for low frequency emphasis. The parallel synthesizer has 2 main advantages :

(a) The noise propagates additively as against multiplicatively in serial one.

(b) It is possible to produce consonant spectra accurately through independent formant amplitude controls. As far as the hardware realization is concerned, the overall control logic will become more complicated. Individual gain control will increase the hardware to a great extent. Hence the trend has been to use serial synthesizer. The schematic diagram of a serial synthesizer is shown in Fig. 3.1. The hardware itself could use serial or parallel arithmetic for the 2 pole filter shown in figure A.1 , representing a difference equation

$$Y = X + a\ (X - YZ^{-1}) + b\ (X - YZ^{-2}) \tag{3.1}$$

or a 2-zero filter representing a difference equation

$$Y = X + cXZ^{-1} + dXZ^{-2} \tag{3.2}$$

These equations arrive directly from the transfer function given by eqns.(2.1), (2.2) for 2-pole and 2-zero filters. The relations are given in the sub-section 3.3.3. Parallel multiplication would increase the hardware enormously (n bit x n bit parallel multiplier will require n - 1 n-bit adders instead of 1). Parallel addition or subtraction would not help if serial multiplier is used. Hence serial arithmetic is chosen in this project.

3.2 INPUT OUTPUT SPECIFICATIONS :-

This section discusses the noise related to word size and filter sequences and concludes with the input output word size and form (serial or parallel, digital or analog). For a discussion on the effect of finite word length, refer to Appendix C .

3.2.1 Effect of Filter Sequences on Signal to Noise Ratio :-

In the cascade combination of 2-pole filters, even though theoretically the output should be unaffected by the interchange of filters, in practice it is always affected due to finite register lengths (word size). Suppose a filter with small coefficient is followed by another with moderate valued coefficient, the former one is likely to give underflow. This output fed to the later one will consequently give highly distorted output. On the otherhand, if the sequence is changed, the first filter will now raise the level of output to a great extent and the next set of coefficients of the cascaded filter will not give underflow. Hence the error introduced in the output will be smaller than before. This shows the importance of having particular sequence of filters and larger register length.

3.2.2 The Word Size :- During the error analysis of 2 sequences of 5-pole digital filters (formants), Gold and Rabiner [10] found out, for synthesis of vowels, the register lengths required at every stage and signal to noise ratio in bits. This required about 15-bit register length (word size) with S/N = 4.5 bits. Also the noise variance was found tobe minimum for an input level of 15 bits. If the number of poles increases, the situation worsens. Hence the word size has tobe still increased. The word size was decided tobe 24 bits by Jackson [2] to account for all truncation, round-off, quantization errors etc. This is the word size also adopted in our project.

3.2.3 The Filter Coefficients :- The input to the synthesizer is a set of 20 coefficients (2 for each filter) tobe available in parallel either from an on-line computer or manually through switches. The other inputs required are pitch, voiced path gain, frication gain all from an on-line real time computer. The actual range of coefficients is between ± 2 but they are weighted so as to perform integer multiplication since outputs of the pulse generator and the frication generator are integers with magnitude less than 2^{23}.

3.2.4 The Output :- The output of the synthesizer is available as a 16 - bit digital output as sum of upper and lower paths. A 12-bit analog speech output can also be obtained. For testing as a digital filter, outputs (and intermediate sums also) of all the 10 filters are available at point 8 of the block diagram of Fig. 3.2 in a serial mode.

3.3 TIME SHARING AND HARDWARE SCHEMATIC :-

This section discusses the feasibility of timesharing with TTL's, describes functions of various blocks of the implemented hardware and illustrates the time sharing and implementation of 2-pole and 2-zero filters.

3.3.1 Time Sharing Feasibility :- The basic principle behind the hardware implementation is the time-sharing of a single arithmetic unit among all the 2-pole and 2-zero filters corresponding to eqns. (3.1) and (3.2) respectively. [11, pp. 210]. In order to realize a 2-pole filter, the two additions, two subtractions and two multiplications are required for each output sample. Medium speed TTL IC's can do about 25 times these operations in the period between output samples (10 KHz sampling rate, IC's operating at 5 MHz rate). Hence the timesharing concept has practical significance in the synthesizer. By providing storage for delayed outputs and dynamically controlling the coefficients, the inputs to filters and the output paths the entire synthesizer can be served by a single arithmetic unit comprising of an adder, a subtractor and a multipler.

3.3.2 Schematic Block Diagram of the Hardware : A schematic block diagram of the hardware for the synthesizer is shown in Fig. 3.2. Blocks I,II,III form an adder with a 1-word wide (24 bits) shift register and a one-out-of-two logic switch controlling the inputs. Blocks V,VI,VII are similar to the above ones except that block V is a one-out-of-three switch and block VII does the subtraction instead of addition. The serial multiplier block VIII has 16-bit parallel coefficients input which is multiplied serially by the output 6 of block VII. Block IX is similar to block V and block X is a 480-bit MOS static shift register, giving a delay of 20 variables for 24-bits per variable. These blocks form the arithmetic unit. The serial output of every filter is available on line 8 and is fed to the output unit consisting of blocks XI,XII,XIII and XIV. Here the outputs of the voiced (upper)path and the fricative (lower) path are added to give the synthesizer output parallely on bus 16 or serially on line 12. Blocks XV and XVI are for D/A conversion for analog speech output. Block XV stores the digital output till the next output sample appears on bus 16. Block IV is the program counter which essentially consists of a counter of 480 followed by a decoder logic to give out various control signals to different blocks, whereever required.

3.3.3 Signal - Flow Pattern :- Table 3.1 shows the signal flow pattern of the synthesizer pertaining to the block diagram of Fig. 3.2. The various notations used are :

X_i = Input to ith 2-pole/2-zero filter.

Y_i = Output of ith 2-pole/2-zero filter.

b_i = First coefficient of ith 2-pole filter.

= e^{-2aT}.

a_i = Second coefficient of ith 2-pole filter.

= $-2\ e^{-aT} \cos b\ T$.

d_i = First coefficient of ith 2-zero filter.

= e^{-2aT}.

c_i = Second coefficient of ith 2-zero filter.

= $-2\ e^{-aT} \cos bT$.

Only the data on the data-lines relevant to the arithmetic unit are tabulated. The output Y_7 is stored in accumulator XIII as a partial sum of the synthesizer output. During the first cycle, output Y_{10} of the lower path is added to this sum to give the synthesizer output.

The Table 3.1 shows how exactly the time-sharing concept is used in the synthesizer hardware. To use the hardware as a bank of 10 digital filters only lines 1 and 4 are to be activated with input straight from the input line every alternate cycles.

As the block diagram of Fig. 3.2 reveals, there is only one input line available as an input to the filters of the synthesizer. This line should be logically connected to either or both the outputs of the pulse generator and the frication generator.

Chapter 4

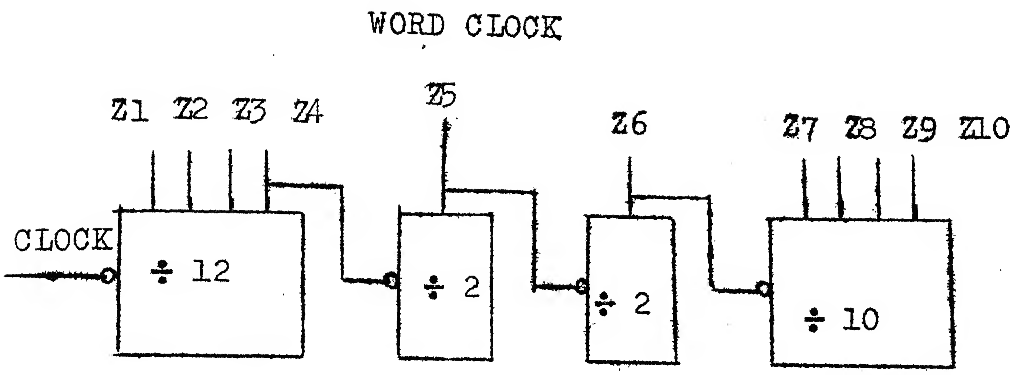
HARDWARE DESIGN CONSIDERATIONS

In this chapter we discuss the design aspects of various blocks explained in the last chapter. The adder and the subtractor blocks are designed together. Since we have decided to use the serial realization, one word length requires one cycle, the terms word and cycle are therefore, used interchangeably.

4.1 THE PROGRAM COUNTER :-

Since the complete iteration of addition, subtraction and multiplication for all filters requires 480 bits, we use a 480-state program counter for controlling the arithmetic operations. This counter is made by cascading a divide-by-twelve counter to a pair of flip flops and a decade counter. (480 = 12 x 2 x 2 x 10) as shown in Fig. 4.1 . This particular structure is convenient to use since a word clock can directly be obtained after a division by 24 (= 12 x 2), as indicated in the diagram.

Indicating the 10 bits of the counter by Z_1, $Z_2 \ldots Z_{10}$ we observe that bits can be split into two groups: Z_1 to Z_5 pertaining to the bits of a word, while Z_6 to Z_{10} give the



THE PROGRAM COUNTER

Fig. 4.1

word count. The control signals are also grouped in two similar categories, the first in terms of bits common to all words and the second common to all bits of a word. Hence these can be obtained in terms of Z_1 to Z_5 and Z_6 to Z_{10} respectively. In this way the number of boolean variables involved in any boolean expression would atmost be five at a time, and as such Karnaugh map minimization with an inclusion of don't care terms can be carried out very easily. Note that a counter of 480 could also be obtained by using only 9 flip flops (Z_1 to Z_9 since $2^9 = 512 > 480$), but that would lead to a much more involved scheme for decoding the signals.

4.2 THE ADDER (BLOCKS I,II,III) AND THE SUBTRACTOR (BLOCKS V,VI,VII) :-

We will refer to the Adder by the suffix U, and to the subtractor by the suffix V. The overall hardware required for both these units are almost similar. The schematic diagram is shown in the Fig. 4.2. The various functions performed by these blocks are described below :

The adder does 2 additions described by eqns. (3.1) and (3.2) in 2 cycles. In even cycles, input X_i of the filter is added to b_i $(X_i - Y_i Z^{-2})$ for 2-pole resonators and to $d_i X_i Z^{-2}$ for 2-zero antiresonators and this partial sum is recirculated through a 24-bit SR when the control signal CU2 is high and is added to a_i $(X_i - T_i\ Z^{-1})$ or $c_i X_i Z^{-1}$ in the odd cycles to give the filter output Y_i.

This output acts as an input to the next cascaded filter except for cycles 1 and 15. when fresh inputs from pulse and noise generator (PNG) for upper and lower path respectively are gated through CU1. The third gate is not used in the adder and hence CU3 = 0. The output of the delay flip-flop D is inhibited at the beginning of each cycle by GU, since the carry input is always zero for the addition of lsb. A gated full adder is used to gate the two inputs Addend and Carry. Since 2's complement addition is performed, the sign bit is also added just like number bits. Overflow can occur only when both the addend and the augend have the same sign and the occurrence is indicated when the sign bit of the sum is not the same as that of the addend or augend. Hence the logic for overflow becomes

$$UOVFLW = \overline{\overline{CU4 \cdot AU1 \cdot AU2 \cdot \overline{U}} \cdot \overline{CU4 \cdot \overline{AU1} \cdot \overline{AU2} \cdot U}} \qquad (4.1)$$

The description of the control signals as well as their boolean expressions obtained after NAND-NOR minimization are given below :

GU = LOW for first bit of all cycles

$$= \overline{\overline{Z1 + Z2} \cdot \overline{Z3} \cdot \overline{Z4 + Z5}} \qquad (4.2)$$

CU = HIGH = 1

CU1= HIGH for cycles 1 and 15 = $\overline{CU2}$

CU2= HIGH for all cycles except 1 and 15

$$= \overline{\overline{Z10} \cdot (\overline{\overline{\overline{Z6} \cdot \overline{Z7} \cdot \overline{Z8} \cdot \overline{Z9}} \cdot \overline{\overline{Z6} \cdot Z7 \cdot Z8 \cdot Z9}})}$$

CU3= LOW = 0

CU4 = HIGH for bit 24 of all cycles.

$$= \overline{\overline{Z1 \cdot Z3 \cdot Z4 \cdot Z5}} \qquad (4.4)$$

The subtractor is required only for 2-pole filters, since the subtrahend is the same for both cycles for each filter, the input X_i is recirculated through control

CV3 for all even cycles. During the 2-zero filter implementation, the 24 bit SR is used to give 2-word delay to the input X_i to 2-zero filter which is then tobe passed to the delay line (480 bit SR) and hence its entry to the subtractor is inhibited by CV. The subtraction is performed by adding 1's complement of minuend together with a 1 at lsb. The carry input is made equal to 1 for the beginning of each cycle by means of the control signal GCV.

The various controls are described below :

GV = HIGH = 1

GCV = LOW for first bit of all cycles = GV

CV = LOW for cycles 1,14,15,20

$$= CU2 \cdot \overline{Z6 \cdot \overline{(\overline{Z7 \cdot Z10} \cdot \overline{\overline{Z7} . Z8 \cdot Z9})}} \qquad (4.5)$$

CV1 = HIGH for cycles 1 and 15 = CU1

CV2 = HIGH for all odd cycles except 1 and 15

$$= \overline{CU1 + Z6} \qquad (4.6)$$

CV3 = HIGH for even cycles = Z6 (4.7)

CV4 = HIGH for last bit of all cycles = CU4

$$VOVFLW = \overline{\overline{\overline{CV4}.AV2.AV1.\overline{V}} \cdot \overline{\overline{AV2}.\overline{AV1}.V.\overline{CV4}}} \qquad (4.8)$$

In the manual mode, when every filter has separate inputs (they no longer remain cascaded) CU1,CV1 gate the inputs every odd cycles and CU2 circulates the partial sum and CV3 recirculates the input every even cycle. Other input gates are not required. Hence, for manual mode

$$CU1 = CV1 = \text{HIGH for odd cycles} = \overline{Z6} \qquad (4.9)$$

$$CU2 = CV3 = \text{HIGH for even cycles} = Z6 \qquad (4.10)$$

$$CV2 = CU3 = 0 \qquad (4.11)$$

Other control signals remain the same.

4.3 MULTIPLIER (BLOCK VIII) :-

The block diagram of the multiplier is shown in Fig. 4.3. Since only 16-bit coefficients are available as multiplicands the remaining 8 m.s. bits are made the same as the sign bit (16th bit) to have 2's complement representation. The multiplication is performed either by shifting the partial product to right and adding the multiplicand if corresponding multiplier bit is 1, or by shifting the multiplicand, gated through the multiplier bit, to the left and adding it to the partial product. The former scheme is used here. The partial product is stored in the 24-bit delay register and shifting is performed by gating every bit through CW to the next ls bit of the 24-bit adder. CW inhibits the contents of the Delay Register for the first bit of every cycle. The multiplicand is gated through multiplier bit for addition. The product is obtained at the lsb sum point of the 24-bit adder. The multiplicand coefficient can be obtained either manually by setting the 2 sets of 16 switches on the panel or from an on-line computer as a 16 bit parallel output. AOI gates are used for the former purpose. Every multiplicand stays in the register for a complete word time and changes next. Therefore, a word clock WCL is given to the clock input of the multiplicand register. The multiplier is fabricated on 6 cards, each one containing a 4 bit multiplier section. The controls are described as below.

CW =: LOW for first bit of every cycle = GV

WCL = CL/24 = Z5

For computer input X = 1, Y = 0, For manual input

$$Y = \overline{Z6} \text{ and } X = Z6 \qquad (4.12)$$

4.4 THE 480 BIT DELAY REGISTER (BLOCK IX, X) :-

This register consists of a 456 bit MOS/LSI static shift register cascaded to a 24 bit SR as shown in Fig. 4.4. The inputs to this register are :

(1) Outputs of the 2-pole filters gated by control signal CX2,

(2) Inputs of the 2-zero filters gated by control signal CX1 and

(3) Delayed outputs or inputs of the 2-pole/2-zero filters respectively gated by control signal CX3. As the data passes from one end to the other, after 480 bits, the inputs are delayed by one sampling period. The sequence in which the output comes out is (starting from first cycle to 20th cycle).

$$X_{10}Z^{-1}, Y_1Z^{-2}, Y_1Z^{-1}, Y_2Z^{-2}, Y_2Z^{-1}, Y_3Z^{-2}, Y_3Z^{-1}, Y_4Z^{-2}, Y_4Z^{-1}, Y_5Z^{-2}, Y_5Z^{-1}, Y_6Z^{-2}, Y_6Z^{-1}, X_7Z^{-2}, X_7Z^{-1}, Y_8Z^{-2}, Y_8Z^{-1}, Y_9Z^{-2}, Y_9Z^{-1}, X_{10}Z^{-2}, X_{10}Z^{-1} \quad (4.13)$$

The control signals are described below :

CX1 = HIGH for cycles 1 and 15 = CU1 (4.14)

CX2 = HIGH for all odd cycles except 1 and 15 = CV2 (4.15)

CX3 = HIGH for even cycles = CV 3 (4.16)

The maximum sampling frequency is limited by this register. Since usual MOS/LSI static SR's can operate at a maximum frequency of 1 MHz, the sampling frequency can atmost be 1/480 MHz = 2 KHz. To increase the sampling frequency, medium speed TTL registers should be used. But then one has to pay in terms of higher cost and higher power dissipation.

4.5 THE OUTPUT UNIT (BLOCKS XI - XV) :-

This unit shown in Fig. 4.5 adds the outputs of the upper and the lower path of the synthesizer to give a 16-bit parallel digital output as well as an analog speech output. The signal CY1 passes the output of line 8 of the upper path of the synthesizer to the 24-bit SR in 15th cycle and the sum of this output and the output of the. lower path in the first cycle. The 24-bit SR holds the sum (synthesizer output) from 2nd to 14th cycle. This sum should be passed to the parallel in parallel out 16-bit SR sometime between these cycles. To reduce the hardware, the sum is passed in the 6th cycle, thus the analog output of any sample is available between 7th and 6th cycles.

The control signals are described below :

CY1 = HIGH for cycle 1 and LOW for cycle 15

$= \overline{\overline{Z6}.Z7.Z8.Z9}$ (available as a part of CU2) (4.17)

CY2 = HIGH for cycle 1 = $\overline{\overline{\overline{Z6}.\overline{Z7}.\overline{Z8}.\overline{Z9}} + Z10}$ (4.18)

GY = LOW for first bit of every cycle = GU (4.19)

YCL1 = CLOCK for cycles 1 and 15 = $\overline{\overline{CL} + CU2}$ (4.20)

YCL2 = a pulse between 7th and 6th cycle = $\overline{Z7+\overline{Z8}}$ (4.21)

In the manual mode, the output of 2-zero filter of the upper path is passed to the register in 15th cycle or the output of the last 2-pole filter is passed in the 19th cycle. The control signals change as follows :

$CY1_{mz}$ = LOW in 15th cycle = $\overline{\overline{Z6}.Z7.Z8.Z9}$ (2-zero filter) (4.22

$CY1_{mp}$ = LOW in 19th cycle = $\overline{\overline{Z6}.Z10}$ (2-pole filter) (4.23

$YCL1 = \overline{\overline{CL} + CY1}$ (4.24

4.6 DECODING SCHEME :-

All the control signals described in sections 4.2 to 4.5 are decoded with NAND-NOR minimization. The intermediate outputs are used to obtain further signals so as to minimize further gating. The decoding scheme, shwon in Fig. 4.6 was realized along with the program counter on one card.

Chapter 5

ASSEMBLY AND TESTING OF THE PRINTED CIRCUIT CARDS

This chapter briefly gives the description of card assembly as well as the results of tests performed. The order of description corresponds to the numbering of the cards. SN7400 series of TTL's and static MOS/LSI has been used for design purposes. Since the later chips were not available, the assembly and testing of Delay Register Card could not be done.

5.1 THE ADDER/SUBTRACTOR :-

5.1.1 Assembly :- Since both of the units have the same structure, only one printed circuit was designed for both. For the adder assembled on card 'A' lines U(pin 14) and Σ(pin 20) were short-circuited, and lines AU1 (pin 9) and Q_{24} (pin 4) were disconnected. For the subtractor assembled on card 'B' the later pair was short-circuited and the former one was open-circuited. The IC chips required on each of these cards are :

IC number	Quantity	IC number	Quantity
SN 7491	3	SN 7420	1
SN 7473	1	SN 7480	1
SN 7410	1	SN 7400	1

5.1.2 Testing :- These blocks were tested for 24-bit word. The printed cards had input

$PNG = \overline{Z5} = 000FFF$ (= 4096) and $AU2 = \overline{\overline{Z4}.\overline{Z5}} = FFFF00$ (-64)

The partial sequence of the output was

U = 000 C7F, 000CBF, 000CFF, 000D3F, 000D7F, 000DBF, 000DFF →

→ 000 E3F, 000E7F, 000EBF, 000EFF, 000F3F, 000F7F, 000FBF →

(First)

5.2 THE MULTIPLIER :-

5.2.1 Assembly :- The 24-bit multiplier consisting of 6 cards C to H has a modular design such that each card separately acts as a 4-bit multiplier. Since the coefficient has only 16 bits, multiplicand of the last 2-cards (G,H) corresponding to msb, are all 0's or all 1's according as the coefficient is positive or negative respectively. This can be done by connecting all the multiplicand bits (line WCL on the card) to the sign bit of the coefficient which is available on card F. This saves 3 IC's (two AOI gates and one 4-bit SR) on both of these cards. The IC's requirements of the cards are shown below :

IC Number	Quantity	IC Number	Quantity
SN 7495	2	SN 7408	1
SN 7483	1	SN 7402	1
SN 7450	2		

(Cards G,H don't need the SN 7450's and one SN 7495)

5.2.2 Testing :- Each card was tested separately. 8 multiplicands were generated by treating the Multiplicand Register as a Mobius counter with the sequence -2,-4,-0,0,1,3,7,-1. The 3 multipliers used were 0,-1; -4; +3. As expected, the product sequences in the 3 cases were

(1) 0,-1: 0 '4 0 0 0 -3 0 1
(2) +3: -6 4^{τ}-0 0 3 -7^{τ} 5^{τ} -3
(3) -4: -0^{τ} 0^{τ} 0 0 -4 4^{τ} 4^{τ} 4

$^{\tau}$ denote overflow conditions.

5.3 THE PROGRAM COUNTER AND DECODING CIRCUIT :-

Since the decoding circuitry was small enough and was directly related to the program counter, the two were combined in card I, the input to which is a clock and the outputs are the control signals for the arithmetic unit and the output. The IC's required are shown below :

IC Number	Quantity	IC Number	Quantity
SN 7400	3	SN 7420	2
SN 7410	1	SN 7402	2
SN 7492	1	SN 7473	1
SN 7490	1		

The printed card was tested for all signals and was found to work satisfactorily.

5.4 THE OUTPUT UNIT :-

The digital output unit consisting of one-out-of-two switch, a 24-bit accumulator/shift register and a gated adder was mounted on card 'J'. The input is the filter output on line 8 and a 16-bit parallel digital output is available. One of the serial in, parallel out 8 bit SR wasn't mounted on this card since this would outnumber the maximum number of lines on a card. Instead, the SR could be mounted on the D/A converter card. The IC's required are shown below :

IC Number	Quantity	IC Number	Quantity
SN 7400	2	SN 7480	1
SN 7473	1	SN 7491	1
SN 74164	1		

This card was also tested for pre-determined input and was found to work satisfactorily.

5.5 OVERFLOW FLIP-FLOP AND DRIVERS :-

Since the fan-out of any gate is 10, buffering is required to drive more number of inputs. This was done in card K which also has 2-flip-flops to indicate overflow in adders of cards 'A'and'B'.

The IC's required are :

IC Number	Quantity	IC Number	Quantity
SN 7400	3	SN 7473	1

The inverters and flip-flops were tested for both HIGH and LOW inputs and were found to give appropriate outputs.

Chapter 6

CONCLUSION

The digital filter implementation is valid only for integer coefficients. If the coefficients are fractions, only sign magnitude multiplication is convenient [see Appendix D]. This will introduce a one word delay in multiplication. Since other operations are in 2's complement form, 2's complementor will be required before and after the multiplier. This changes the control signals and the interconnections remarkably.

The equipment could not be thoroughly tested as a speech synthesizer because of the non-availability of software required to generate coefficients and Gains and Pitch period corresponding to a given phoneme train and formant motion. Different rules of the synthesis strategies have tobe implemented by software and a library of key words is required.

Suggested Modifications in Hardware :-

A 320 bit RAM to store one set of 16-bit coefficients will facilitate examining the synthesis of vowels.

A combinatorial selector network to select any 12 consecutive output bits for D/A conversion will be useful in noise studies - where one can analyse the noisy bits of the synthesizer output.

Appendix - A

Z - TRANSFORMS

Just as Laplace transform describes the response of continuous filters, the sampled data (Z) transform describes the behaviour of digital filters. The various transformations from S -domain to Z-domain are (1) the standard Z-transform (2) the bilinear Z-transform (3) the matched Z-transform. Their applicability is discussed below :

A.1 THE STANDARD Z - TRANSFORM : -

This transform yields a digital filter with impulse response equal to the sampled impulse response of the continuous filter (hence also referred to as impulse invariant transformation). The transform is defined as

$$Z\left(\frac{K}{s+a}\right) = \frac{K}{1 - e^{-aT} Z^{-1}} \qquad (A.1)$$

and $$Z\left(\sum \frac{K_i}{s + a_i}\right) = \sum \frac{K_i}{1 - e^{-a_i T} Z^{-1}} \qquad (A.2)$$

The complex poles are transformed as

$$Z\left(\frac{K}{(s+a)^2 + b^2}\right) = \frac{K(e^{-aT} \sin bT\, Z^{-1} - e^{-2aT} Z^{-2})}{1 - 2\, e^{-aT} \cos bT\, Z^{-1} + e^{-2aT} Z^{-2}} \qquad (A.3)$$

where T = Sampling period, and

$Z^{-1} = e^{-sT}$ = unit delay variable.

This representation gives excellent results for lowpass and bandpass filters. But it gives rise to folding effects since the digital filters frequency response is repetetive with a periodic frequency equal to sampling frequency ($f_r = 1/T$). This can be shown by writing $Z = e^{jwT}$ and $r = e^{-aT}$ in Denominator of A-3 whose magnitude then becomes :

$$|\text{Denominator}| = [1+r^2 - 2r \cos (w-b)T]^{\frac{1}{2}} \text{ x}$$
$$[1+r^2 - 2r \cos (w+b)T]^{\frac{1}{2}} \qquad (A.4)$$

A.2 THE BILINEAR Z - TRANSFORM : -

The folding effects of the standard Z - transforms are avoided in the bilinear Z - transform which is an algebraic transform defined by

$$Z(s) = \frac{2}{T} \cdot \frac{1 - Z^{-1}}{1 + Z^{-1}} \qquad (A.5)$$

Being algebraic in nature, either parallel or serial realization is possible in this transform. The price paid is in terms of frequency warping near the critical frequencies, the warping being given by

$$f_c = \frac{1}{\pi T} \tan \pi f_z T \qquad (A.6)$$

where f_c = critical frequency

and, f_z = frequency in z-domain.

This makes the transform useless for HIGH-Q bandpass filters.

A.3 THE MATCHED Z - TRANSFORM : -

This transform yields a series realization of digital filters. The poles and zeros of the digital filter generated by this transform match to those of a continuous function. The mapping is given by

$$s \rightarrow e^{sT} = Z$$

The poles or zeros transform according to

$$Z(s + u) = 1 - e^{-uT} Z^{-1} \qquad (A.7)$$

$$Z\pi(s+u_i) = \pi Z(s+u_i) = \pi(1 - e^{-u_i T} Z^{-1}) \qquad (A.8)$$

$$Z((s+u)^2+v^2) = 1 - 2e^{-uT} \cos vT\, Z^{-1} + e^{-2uT} Z^{-2} \qquad (A.9)$$

The transfer function of complex poles in z-domain with unity gain at zero frequency is realized in Fig. A-1. (Here $r = e^{-uT}$ and $b = v$).

One can note that the poles of the TF are the same as those derived by the standard Z-transform but the zeros may not transform accordingly. For a bandpass design gain scale is a little distorted.

In general, the frequency response of a digital filter can be geometrically obtained from the pole zero diagram as shown in Fig. A.2. The magnitude of the output is proportional to the distance of point P (defined at a frequency w, by 1 $\angle wT$) from the zeros (which is unity in case of poles) and inversely proportional to the distances from the poles.

Appendix B

COEFFICIENT VALUES FOR THE FORMANTS OF PHONEMES AT SAMPLING FREQUENCY OF 1 KHz

PHONEME	TYPE	F1-Hz	BW1-Hz	COEFF.	COEFF.
				A	B
IY	V	270	150	-0.098	0.152
I	V	390	150	-0.600	0.152
E	V	530	150	-0.766	0.152
AE	V	660	150	-0.418	0.152
UH	V	520	150	-0.773	0.152
A	V	730	74	-0.157	0.395
OW	V	570	150	-0.705	0.152
U	V	440	150	-0.725	0.152
OO	V	300	150	-0.241	0.152
ER	V	490	150	-0.778	0.152
W	V	300	50	-0.451	0.534
L	V	380	50	-1.065	0.534
R	V	420	60	-1.202	0.471
Y	V	300	50	-0.451	0.534
B	V	000	100	1.067	0.285
D	V	000	60	1.372	0.471
G	V	000	30	1.656	0.686
M	NV	280	34	-0.303	0.652

Data for the First Formant

.....Appendix B contd......

PHONEME	TYPE	F1-Hz	BW1-Hz	COEFF. A	COEFF. B
N	NV	280	34	-0.303	0.652
NG	NV	280	34	-0.303	0.652
P		000	100	1.067	0.285
T		000	60	1.372	0.471
K		000	20	1.764	0.778
F	F	175	60	0.623	0.471
TH	F	200	60	0.424	0.471
S	F	200	60	0.424	0.471
SH	F	175	60	0.623	0.471
V	FV	175	60	0.623	0.471
THE	FV	200	60	0.424	0.471
Z	FV	200	60	0.424	0.471
ZH	FV	175	60	0.623	0.471

Data for the first formant

....Appendix B contd,.....

PHONEME	TYPE	F2-Hz	BW2-Hz	COEFF. A	COEFF. B
IY	V	2290	150	-0.194	0.152
I	V	1990	150	0.778	0.152
E	V	1840	160	0.392	0.134
AE	V	1720	150	-0.146	0.152
UH	V	1190	150	0.287	0.152
A	V	1090	150	0.658	0.152
OW	V	840	150	0.418	0.152
U	V	1020	150	0.773	0.154
OO	V	870	160	0.501	0.134
ER	V	1350	160	-0.430	0.134
W	V	610	80	-0.932	0.336
L	V	880	160	0.534	0.134
R	V	1300	160	-0.226	0.134
Y	V	2200	220	0.155	0.063
B	V	800	150	0.241	0.152
D	V	1700	100	-0.330	0.285
G	V	2350	100	-0.627	0.285
M	NV	900	34	1.307	0.652

Data for the second formant

....Appendix B contd....

PHONEME	TYPE	F2-Hz	BW2-Hz	COEFF. A	COEFF. B
N	NV	1700	34	-0.499	0.652
NG	NV	2300	34	-0.499	0.652
P		800	80	0.374	0.366
T		1700	60	-0.424	0.471
K		2350	60	-0.806	0.471
F	F	900	100	0.863	0.285
TH	F	1400	80	-0.979	0.366
S	F	1300	80	-0.374	0.366
SH	F	1800	200	0.176	0.081
V	FV	1100	100	0.863	0.285
THE	FV	1600	80	-0.979	0.366
Z	FV	1300	80	-0.374	0.366
ZH	FV	1800	200	0.176	0.081

Data for the Second Formant

Appendix B contd.......

PHONEME	TYPE	F3-Hz	BW3-Hz	COEFF. A	COEFF. B
IY	V	3010	300	0.303	0.023
I	V	2550	220	-0.477	0.063
E	V	2480	220	-0.498	0.063
AE	V	2410	220	-0.424	0.063
UH	V	2390	150	-0.600	0.152
A	V	2440	230	-0.438	0.056
OW	V	2410	230	-0.398	0.056
U	V	2240	180	0.041	0.104
OO	V	2240	180	0.041	0.104
ER	V	1690	200	-0.210	0.081
W	V	2200	300	0.094	0.023
L	V	2575	300	-0.271	0.023
R	V	1600	200	-0.461	0.081
Y	V	3065	400	0.149	0.007
B	V	1750	240	0.000	0.049
D	V	2600	320	-0.217	0.018
G	V	2000	200	0.569	0.081
M	NV	2200	80	0.374	0.366

Data for the Third Formant

Appendix B contd.........

PHONEME	TYPE	F3-Hz	BW3-Hz	COEFF. A	COEFF. B
N	NV	2600	200	-0.461	0.081
NG	NV	2750	200	-0.000	0.081
P		1750	160	0.000	0.134
T		2600	200	-0.461	0.081
K		2000	140	0.830	0.172
F	F	2400	240	-0.358	0.049
TH	F	2200	200	0.176	0.081
S	F	2500	140	-0.830	0.172
SH	F	2000	300	0.304	0.023
V	FV	2400	240	-0.258	0.049
THE	FV	2200	200	0.176	0.081
Z	FV	2500	140	-0.830	0.172
ZH	FV	2000	300	0.304	0.023

Data for the Third Formant

The type column in all tables shows whether the phoneme is voiced, nasal or fricative.

Appendix C

ERRORS DUE TO FINITE WORD LENGTH

Accuracy of a digital filter is limited by the finite word length used. The different sources of errors are (1) Input quantization errors (2) multiplicative round off/truncation errors (3) Errors due to quantization of coefficients and (4) Errors due to overflow arithmetic.

C.1 INPUT QUANTIZATION ERRORS :-

Integer input X with a n-bit word can be represented in 2's complement form as

$$X = -x_n + \sum_{i=n-1}^{0} x_i \, 2^i \qquad (x_i = 0 \text{ or } 1) \qquad \text{(C-1)}$$

The error in this form can be in the lsb (least significant bit) x_0. If round off at x_0 is performed, error will be reduced to half, i.e. $\frac{1}{2}$ in a word of maximum value of 2^{n-1}. In other words, the error as a fraction of the word is

$$E = 2^{2-n} \qquad \text{(C.2)}$$

Thus the error reduces as the word length n increases. For further analysis one is referred to [11, pp. 361].

C.2 ERRORS DUE TO MULTIPLICATION :-

Product quantization is the rounding of the product of 2 already quantized numbers. The product of m bit x n bit fractions will be m+n bits long. If the product is rounded to l bits, the error made will assume certain discrete values [12]. It has been shown that the product quantization error is large compared with other errors. The mean squared error interms of maximum input quantization error E_o is given by

$$E(|e^2|) = \frac{E_o}{3}[E_c + 2^{-(m+n)}] \quad \text{where } E_o=2^{-(l+1)} \qquad (C.3)$$

It should be noted that, if

$$l \text{ is greater than } m + n \qquad (C.4$$

Then no quantization error occurs. For multiplication with fractions, only underflow can occur. If in the integer multiplication, condtion C.4 is not satisfied, there is a possibility of overflow of product. But as long as the product is less than l bits long and if m = n, it has been found out that the product is exact for integers. This is shown in Appendix D.

C.3 ERRORS DUE TO QUANTIZATION OF COEFFICIENT :-

The filter characteristics are altered as a result of coefficient errors due to finite word length. Kaiser [11, pp. 369] has shown that stability of a lowpass filter with sampling period T and distinct poles at $e^{-P_k T}$ is guaranteed. only if the number of bits

$$m_b \geqslant n[1+\log_2(\frac{1}{T})] - \sum_{k=1}^{n} \log_2 P_k \qquad (C.5)$$

Actual word length may have to be larger than this, to ensure acceptable accuracy.

C.4 ERRORS DUE TO OVERFLOW ARITHMETIC :-

Large errors may occur in the output, if as result of overflow in adders of multipliers, output of these units is not kept at their maximum (hard limiting). This may be a very difficult task if serial arithmetic is used.

Appendix D

2's COMPLEMENT INTEGER MULTIPLICATION

D.1 SIGN MAGNITUDE VERSUS 2's COMPLEMENT MULTIPLICATION :-

In the conventional multiplication, the multiplicand and the multiplier both are in sign magnitude form and the product is available with one word delay. It is valid for fractions and there is always truncation (or round off) error and a possibility of underflow. In the new, though unconventional scheme where integer multiplication in sign 2's complement form is performed, the product is immediately available in 2's complement integer form. The serial product comes out without delay as the multiplier is serially fed in. Another advantage of this method is apparent from the fact that no prior knowledge of the sign bit of the multiplier is necessary. The hardware required for this scheme is essentially the same as that required for the conventional scheme, but the product bits considered are different, and sign bits are treated as number bits.

The new scheme operates in the region which gives underflow in fractional multiplication. The advantage of this scheme is that there is no round off or truncation error. But this is nullified by the fact that there is a great possibility of an overflow (in the region in which fractional product has non-zero value). The following is a brief mathematical proof to show that product of 2 negative numbers in 2's complement form yield a positive product.

D.2 PROOF OF VALIDITY OF THE SCHEME :-

The negative n bit multiplicand X and n bit multiplier Y in 2's complement form can be represented as

$$X = (-1)2^n + \sum_{i=0}^{n-1} \bar{x}_i 2^i + 1 \quad \text{where } \bar{x}_i = 1-x_i \qquad (D.1)$$

$$Y = (-1)2^n + \sum_{j=0}^{n-1} \bar{y}_j 2^j + 1 \quad \text{where } \bar{y}_j = 1-y_j \qquad (D.2)$$

then the product will be given by

$$X \cdot Y = 2^{2n} - 2^n \left(\sum_i \bar{x}_i 2^i + \sum_j \bar{y}_j 2^j\right) + \sum_i \bar{x}_i 2^i + \sum_j \bar{y}_j 2^j + \sum_i \sum_j \bar{x}_i \bar{y}_j \, 2^{i+j} - 2^{n+1} + 1$$

where $\sum_k$ denotes summation from $k = 0$ to $k = n-1$.

$$X \cdot Y = 2^{2n} - \sum_i \sum_j (\bar{x}_i + \bar{y}_j) 2^{i+j} + \sum_i \sum_j \bar{x}_i \bar{y}_j \, 2^{i+j} - 2^{n+1} + 1$$

Now putting $\bar{x}_i = 1 - x_i$ and $\bar{y}_j = 1 - y_j$

$$X \cdot Y = 2^{2n} - \sum_i \sum_j 2^{i+j+1} + \sum_i \sum_j (x_i + y_j) 2^{i+j} + \sum_i \sum_j 2^{i+j}$$

$$- \sum_i \sum_j (x_i + y_j) 2^{i+j} + \sum_i \sum_j x_i y_j \, 2^{i+j} - 2^{n+1} + 1$$

$$= 2^{2n} - \sum_i \sum_j 2^{i+j} - 2^{n+1} + 1 + \sum_i \sum_j x_i y_j \, 2^{i+j}$$

$$= \sum_i \sum_j x_i y_j \, 2^{i+j} \quad \left(\sum_i \sum_j 2^{i+j} = (2^n - 1)^2 = 2^{2n} - 2^{n+1} + 1 \right) \qquad (D.3)$$

A similar proof can be given for a product of a positive and a negative integer.

REFERENCES

[1] Flanagan, J.L.: Speech Analysis Synthesis and Perception, N.Y. Academic Press Inc.

[2] Rabiner, L.R.; Jackson, L.B.: A Hardware Realization of a Digital Formant Speech Synthesizer, IEEE Trans. on COMM. Tech., Vol. COM-19, No.6, pp. 1016-1020, Dec. '71.

[3] Golden, R.M.: Digital Filter Synthesis by Sampled Data Transformation, IEEE Trans. on Audio and Electro-acoustics, N.Y., 3 April, 1968.

[4] Rabiner, L.R.: Speech Synthesis by Rule: An Acoustic Domain Approach, M.I.T. Cambridge, Mass. and BTL, Inc., Murray Hill, N.J.

[5] Flanagan, J.L., Coker, C.H., Schafer, R.W., Umeda, N.: Synthetic Voices for Computers, IEEE Spectrum, Vol.7, No.10, pp. 22-45, Oct. '70.

[6] Rabiner, L.R.: Digital Formant Synthesizer for Speech Synthesis Studies, J. of Accou. Soc. of Am., Vol.43, pp. 822-828, April, '68.

[7] Signal Processors and Synthesizers, Rockland Systems Corp. N.Y. 1972, Product Catalog.

[8] Tomlinson, R.S.: An Improved Solid State Terminal Analog Speech Synthesizer, M.S. Thesis, M.S.T. June, '65.

[9] Fant, G.: Speech Analysis and Synthesis, Royal Inst. of Tech., Speech Transmission Lab., Stockholm, Sweden, June, '62.

[10] Gold, B., Rabiner, L.R.: Analysis of Digital and Analog Formant Synthesizers, BTL Inc., N.J.

[11] Rabiner, L.R.: Digital Signal Processing; 621.38043/R113.

[12] Halyo, M. et al.: A discrete Model for Product Quantization Errors in Digital Filters, IEEE Trans. on AU, Vol. AU-19, pp. 255-256, Sept. 1971.

TABLE 3.1 SIGNAL FLOW PATTERN

data line / cycle	1	3	5	6	7	8	9	11
	Input Line	Coeff.	Subtrahend	Multiplier	Product	Output	Delay SR Input	Delay SR Output
1	(INPUT) X_1	c_{10}	X_{10}	$-X_{10}Z^{-1}$	$c_{10}X_{10}Z^{-1}$	Y_{10}	X_{10}	$X_{10}Z^{-1}$
2	$X_1+b_1(X_1-Y_1Z^{-2})$	b_1	X_1	$X_1-Y_1Z^{-2}$	$b_1(X_1-Y_1Z^{-2})$	$X_1+b_1(X_1-Y_1Z^{-2})$	Y_1Z^{-1}	Y_1Z^{-2}
3	X_2	a_1	X_1	$X_1-Y_1Z^{-1}$	$a_1(X_1-Y_1Z^{-1})$	$Y_1 \Rightarrow X_2$	Y_1	Y_1Z^{-1}
4	$X_2+b_2(X_2-Y_2Z^{-2})$	b_2	X_2	$X_2-Y_2Z^{-2}$	$b_2(X_2-Y_2Z^{-2})$	$X_2+b_2(X_2-Y_2Z^{-2})$	Y_2Z^{-1}	Y_2Z^{-2}
5	X_3	a_2	X_2	$X_2-Y_2Z^{-1}$	$a_2(X_2-Y_2Z^{-1})$	$Y_2 \Rightarrow X_3$	Y_2	Y_2Z^{-1}
6	$X_3+b_3(X_3-Y_3Z^{-2})$	b_3	X_3	$X_3-Y_3Z^{-2}$	$b_3(X_3-Y_3Z^{-2})$	$X_3+b_3(X_3-Y_3Z^{-2})$	Y_3Z^{-1}	Y_3Z^{-2}
7	X_4	a_3	X_3	$X_3-Y_3Z^{-1}$	$a_3(X_3-Y_3Z^{-1})$	$Y_3 \Rightarrow X_4$	Y_3	Y_3Z^{-1}
8	$X_4+b_4(X_4-Y_4Z^{-2})$	b_4	X_4	$X_4-Y_4Z^{-2}$	$b_4(X_4-Y_4Z^{-2})$	$X_4+b_4(X_4-Y_4Z^{-2})$	Y_4Z^{-1}	Y_4Z^{-2}
9	X_5	a_4	X_4	$X_4-Y_4Z^{-1}$	$a_4(X_4-Y_4Z^{-1})$	$Y_4 \Rightarrow X_5$	Y_4	Y_4Z^{-1}
10	$X_5+b_5(X_5-Y_5Z^{-2})$	b_5	X_5	$X_5-Y_5Z^{-2}$	$b_5(X_5-Y_5Z^{-2})$	$X_5+b_5(X_5-Y_5Z^{-2})$	Y_5Z^{-1}	Y_5Z^{-2}

TABLE 3.1 CONTD......

data line / cycle	1	3	5	6	7	8	9	11
	Input Line	Coeff.	Subrahend	Multiplier	Product	Output	Delay SR input	Delay SR output
11	X_6	a_5	X_5	$X_5-Y_5Z^{-1}$	$a_5(X_5-Y_5Z^{-1})$	$Y_5 \Rightarrow X_6$	Y_5	Y_5Z^{-1}
12	$X_6+b_6(X_6-Y_6Z^{-2})$	b_6	X_6	$X_6-Y_6Z^{-2}$	$b_6(X_6-Y_6Z^{-2})$	$X_6+b_6(X_6-Y_6Z^{-2}$	Y_6Z^{-1}	Y_6Z^{-2}
13	X_7	a_6	X_6	$X_6-Y_6Z^{-1}$	$a_6(X_6-Y_6Z^{-1})$	$Y_6 \Rightarrow X_7$	Y_6	Y_6Z^{-1}
14	$X_7+d_7X_7Z^{-2}$	$-d_7$	X_7	$-X_7Z^{-2}$	$d_7X_7Z^{-2}$	$X_7+d_7X_7Z^{-2}$	X_7Z^{-1}	X_7Z^{-2}
15	(INPUT) X_8	c_7	X_7	$-X_7Z^{-1}$	$c_7X_7Z^{-1}$	Y_7	X_7	X_7Z^{-1}
16	$X_8+b_8(X_8-Y_8Z^{-2})$	b_8	X_8	$X_8-Y_8Z^{-2}$	$b_8(X_8-Y_8Z^{-2})$	$X_8+b_8(X_8-Y_8Z^{-2}$	Y_8Z^{-1}	Y_8Z^{-2}
17	X_9	a_8	X_8	$X_8-Y_8Z^{-1}$	$a_8(X_8-Y_8Z^{-1})$	$Y_8 \Rightarrow X_9$	Y_8	Y_8Z^{-1}
18	$X_9+b_9(X_9-Y_9Z^{-2})$	b_9	X_9	$X_9-Y_9Z^{-2}$	$b_9(X_9-Y_9Z^{-2})$	$X_9+b_9(X_9-Y_9Z^{-2}$	Y_9Z^{-1}	Y_9Z^{-2}
19	X_{10}	a_9	X_9	$X_9-Y_9Z^{-1}$	$a_9(X_9-Y_9Z^{-1})$	$Y_9 \Rightarrow X_{10}$	Y_9	Y_9Z^{-1}
20	$X_{10}+d_{10}X_{10}Z^{-2}$	$-d_{10}$	X_{10}	$-X_{10}Z^{-2}$	$d_{10}X_{10}Z^{-2}$	$X_{10}+d_{10}X_{10}Z^{-2}$	$X_{10}Z^{-1}$	$X_{10}Z^{-2}$

Table 5.1 Connection Diagram of the Printed Circuit Cards:·

CARD	A	B	C	D	E	F	G	H
Pin	Adder	Subtractor	Multiplier					
1	NC	NC	NC	NC	NC	NC	NC	NC
2	B/2	I/20	NC	NC	NC	NC	NC	NC
3	NC	NC	D/4	E/4	F/4	G/4	H/4	H/5
4	NC	NC	–	C/3	D/3	E/3	F/3	G/3
5	–	–	D/8	E/8	F/8	G/8	H/8	H/3
6	$\overline{CL}$,B/6	$\overline{CL}$,K16	+	+	+	+	+	+
7	CL	CL	CL	CL	CL	CL	CL	CL
8	+	+	A/19	C/5	D/5	E/5	F/5	G/5
9	NC	NC	D/9	I/21	F/9	G/9	H/9	K/1, J/3
10	–	I/16	D/10	C/10	F/10	K/9	H/10	K/11
11	B/11	K/6	–	–	–	–	–	–
12	INP B/12	INP A/12	D/12	K/4	F/12	K/7	NC	NC
13	SW,CU2	SW,CV2	WY3	WY7	WY11	WY15	NC	NC
14	NC	A/20 J/7	WX3	WX7	WX11	WX15	NC	NC
15	NC	NC	D/15	K/3	F/15	K/8	NC	NC
16	K/12	K/10	WX2	WX6	WX10	WX14	NC	NC
17	+	I/4	WY2	WY6	WY10	WY14	NC	NC
18	B/18	C/9	WY1	WY5	WY9	WY13	NC	NC
19	C/8	$\overline{M}_1$	WX1	WX5	WX9	WX13	NC	NC
20	B/14	K/13	WX4	WX8	WX12	WX16	NC	NC
21	NC	C/10	WY4	WY8	WY12	WY16	NC	NC
22	NC	NC	D/22	E/22	F/22	I/18	H/22	A16 Note 2

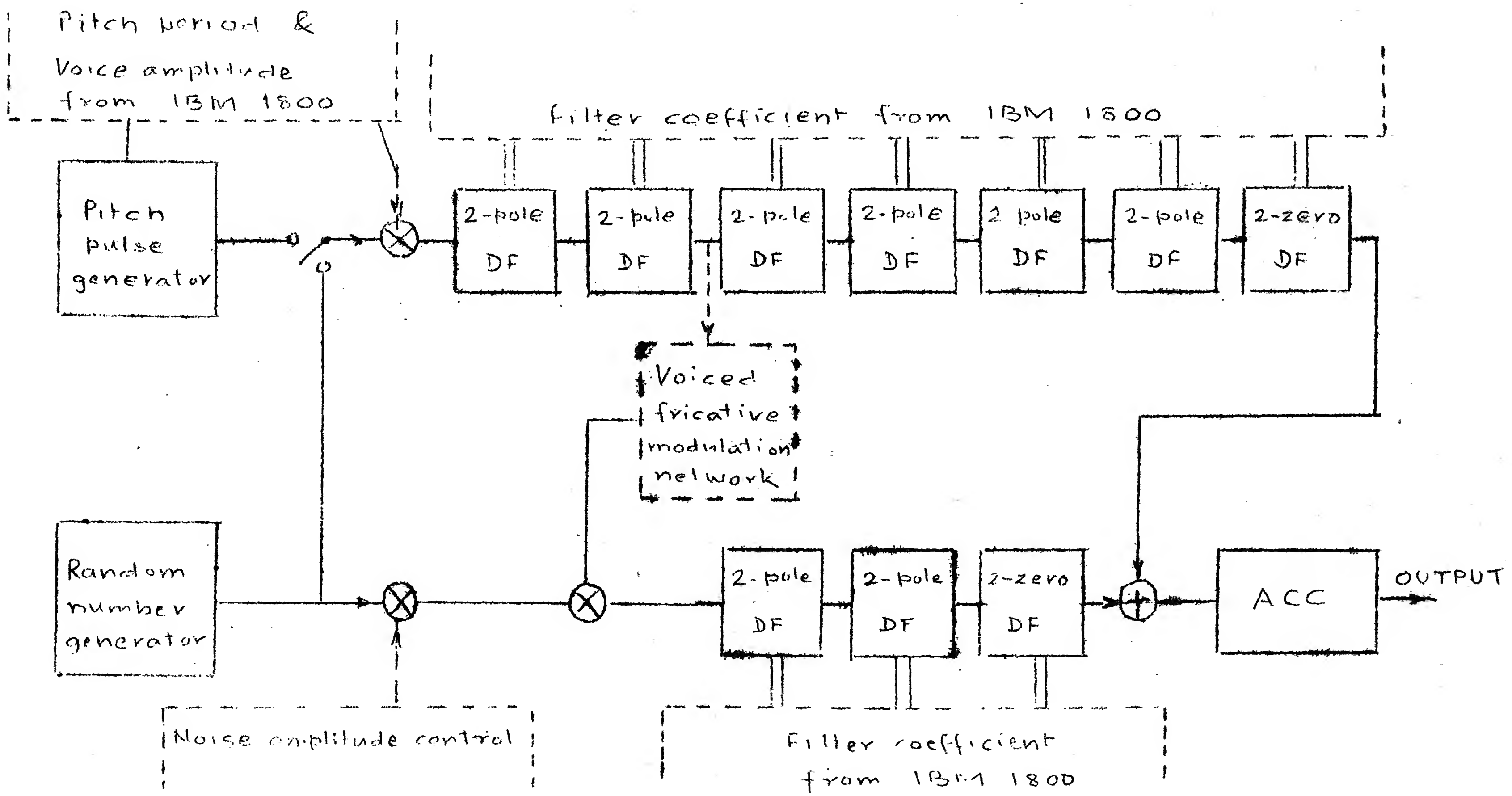
Note: 1 Output at J/5 is input to serial in, Parallel out 8 bit SR with clock = YCL 1 and the output of this SR is connected to J/20.

Note: 2 H/22 is connected to A16 which is input to the register of card F (WX16.X + WY16.Y) internally (point A16 is internal to card F).

Note: 3 I/3 is a clock to the register which is input of D/A converter.

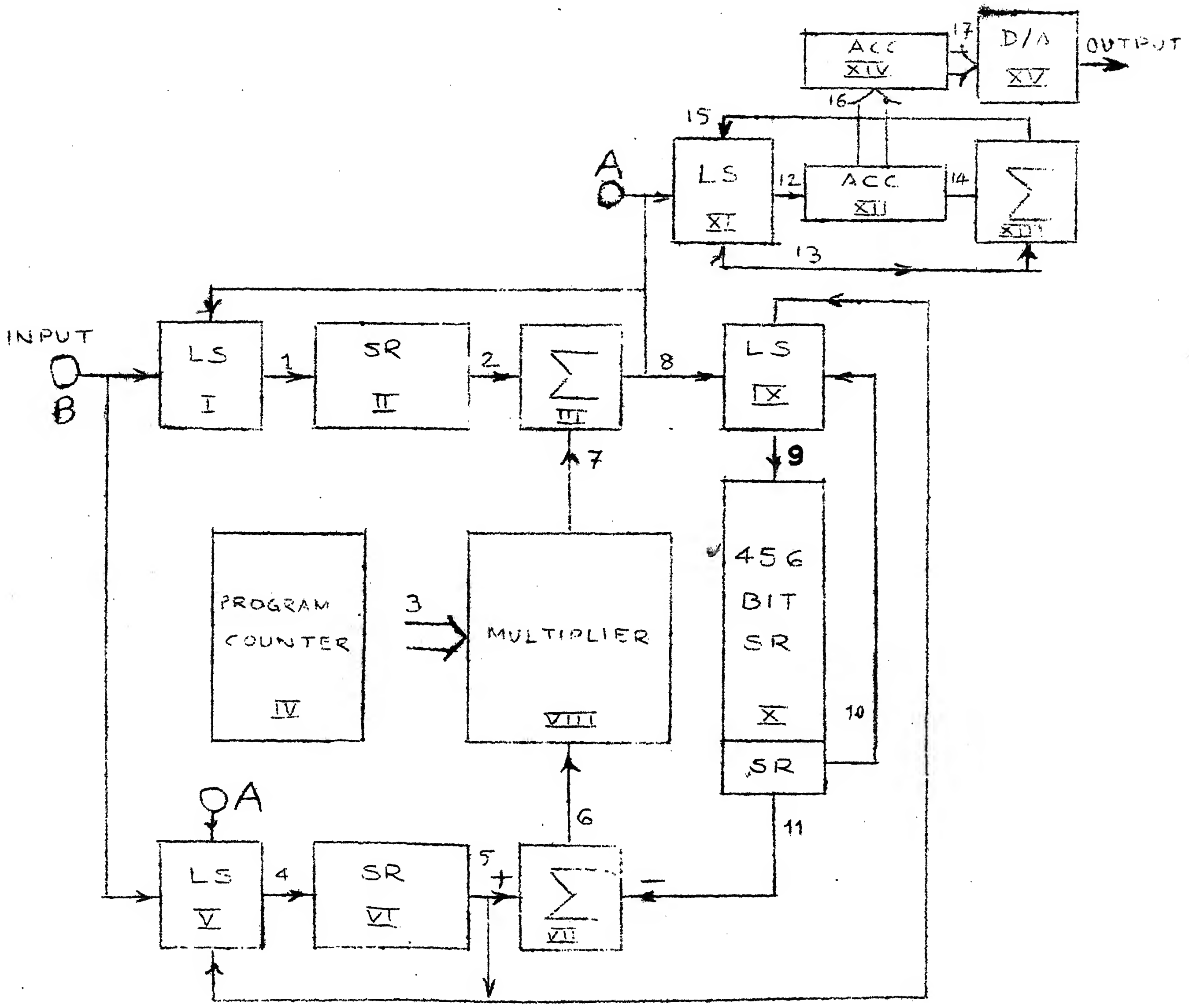
Table 5.1 .... contd......

CARD	I	J	K
Pin	CONTROL	OUTPUT	BUFFER
1	To SW CV2	NC	H29,J/3
2	J/4	NC	I/21
3	Note 3	K/1	D/15
4	B/17	I/2	D/12 and switch out
5	NC	Note 1	Output of SW CU2
6	To SW CU2	-	B/11
7	SPDT,CY1$_{mp}$	B/14	F/12
8	+	Output of SPDT	F/15
9	NC	+	F/10
10	J/21	CL	B/16
11	SPDT,CY1$_{mz}$	NC	H/10
12	NC	Q8	A/16
13	K/20	Q7	B/20
14	-	Q6	+
15	K/21	Q5	CL in
16	B/10	Q4	$\overline{\text{CL}}$ out
17	SW,$\overline{\text{Z6}}$	Q3	LED OVA
18	F/22	Q2	LED OVB
19	CL	Q1	CL out
20	B/2	Note 1	I/13 and µ switch
21	K/2	I/10	I/15
22	Output of switch for YCL1	NC	-



SCHEMATIC DIAGRAM OF SERIAL SPEECH SYNTHESIZER

FIGURE 3.1



Roman numericals are block identification numbers.

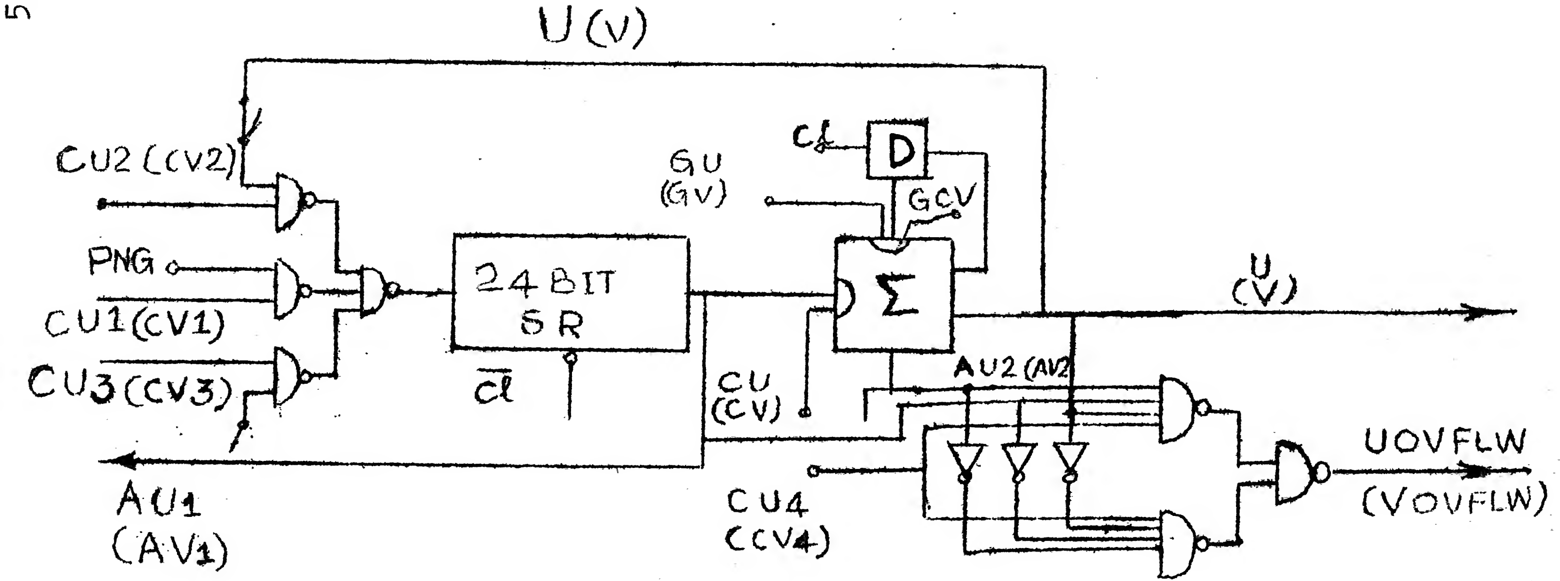
Numbers above lines are data line numbers.

2 POLE FILTERS :- $Y_i = x_i + b_i(x_i - Y_i Z^{-2}) + a_i(x_i - Y_i Z^{-1})$

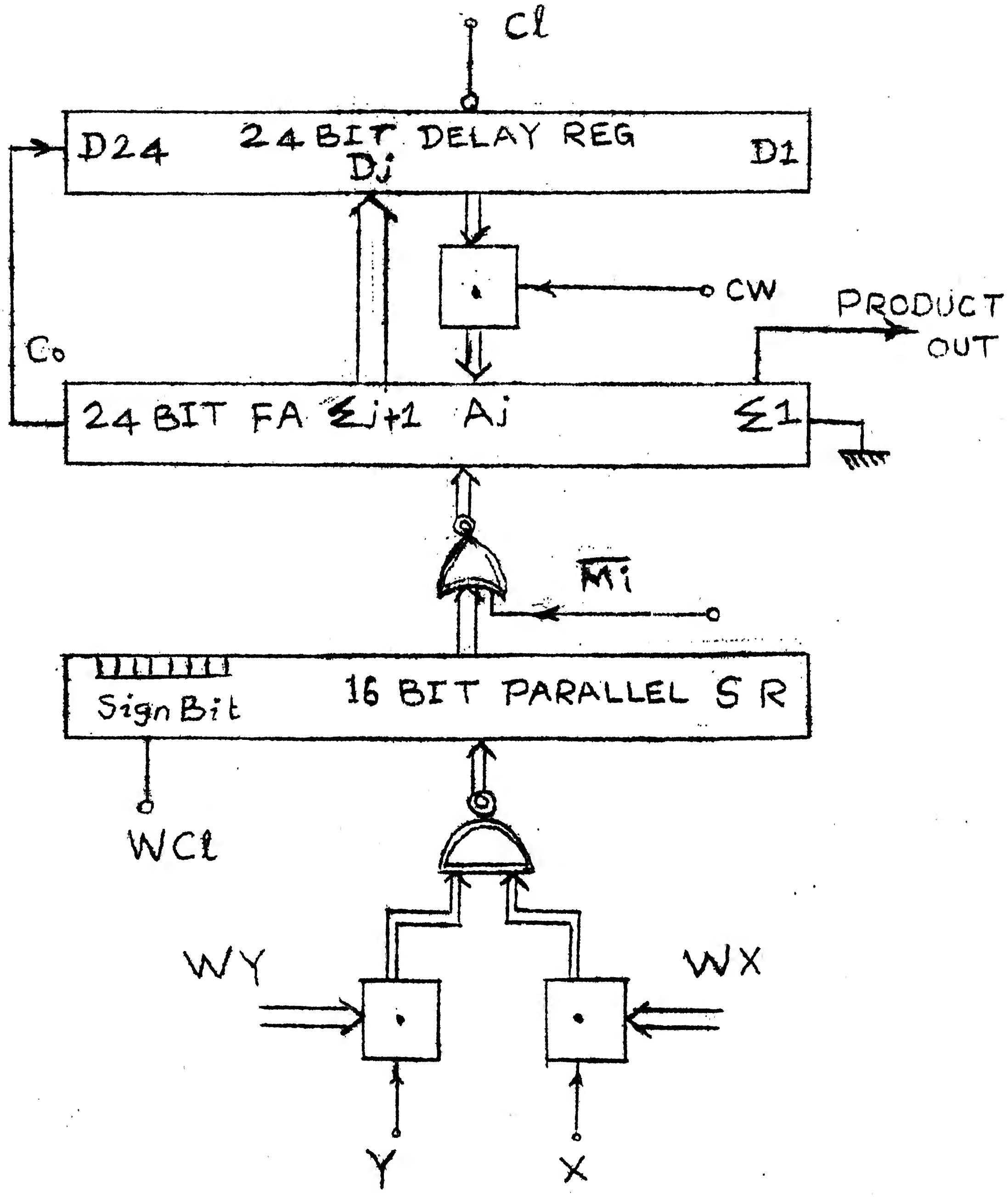
2 ZERO FILTERS :- $Y_i = x_i + d_i X_i Z^{-2} + C_i X_i Z^{-1}$

THE HARDWARE SCHEMATIC

FIGURE 3·2

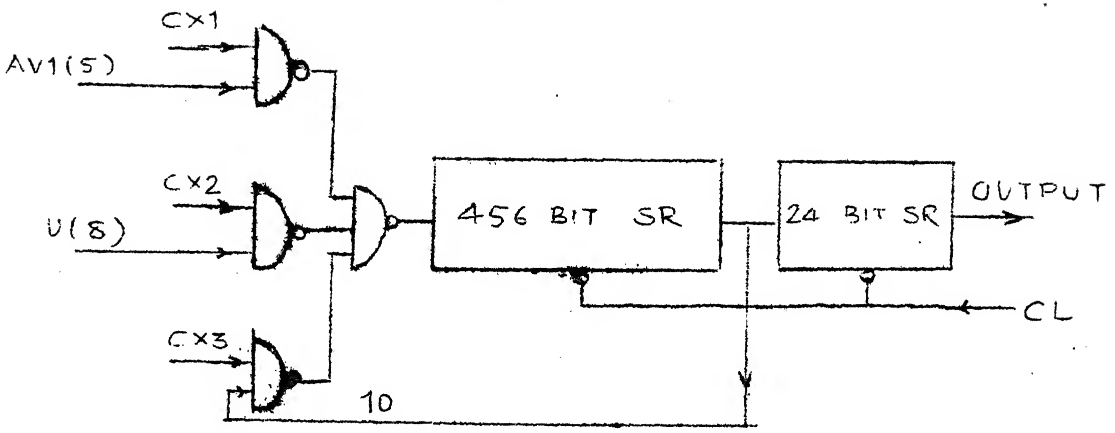


ADDER/SUBTRACTOR
SCHEMATIC
FIG 4·2



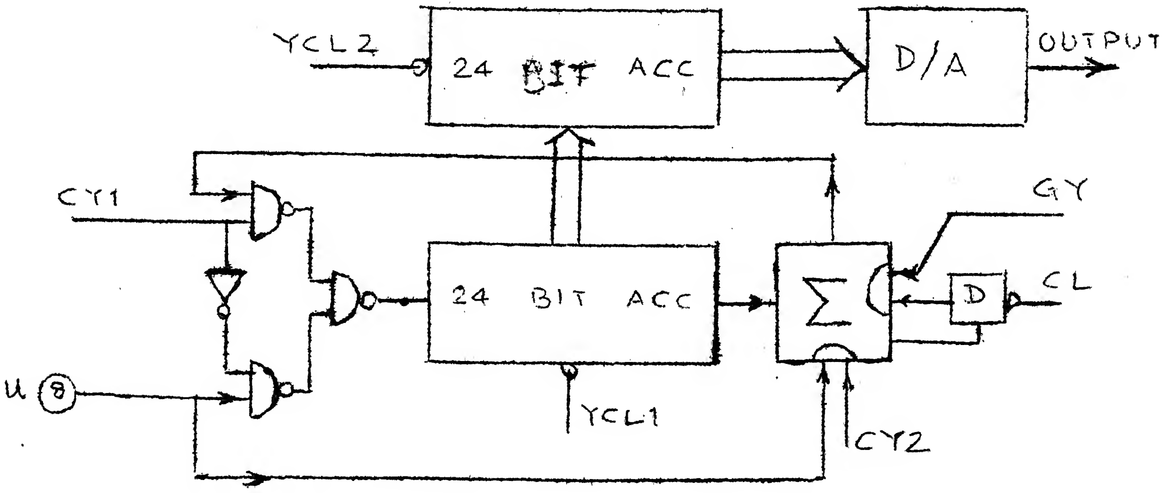
THE MULTIPLIER

FIG 4·3



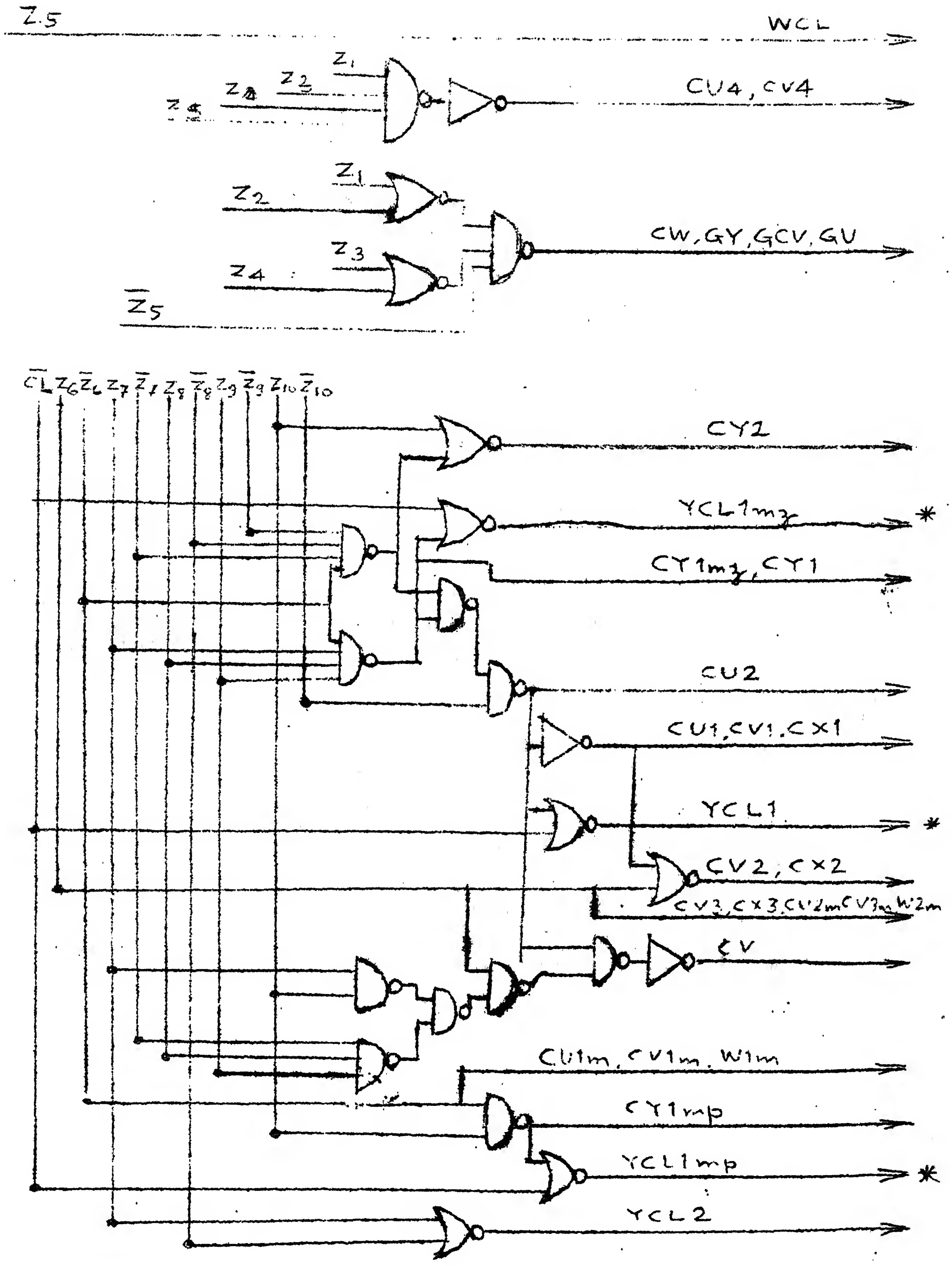
480 BIT DELAY REGISTER

FIGURE 4·4



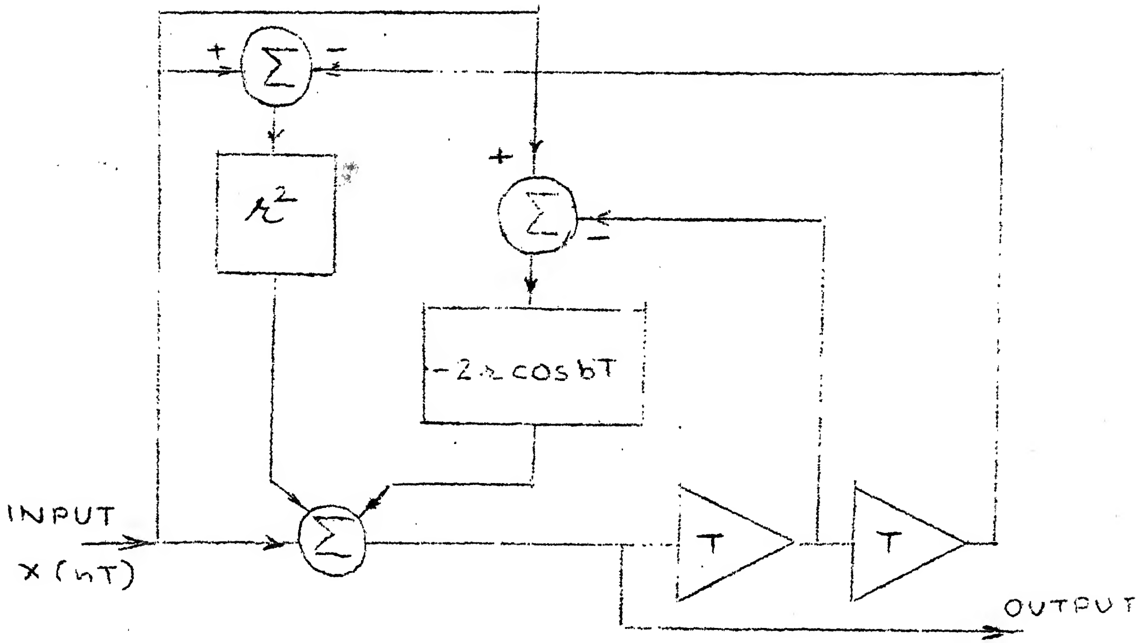
THE OUTPUT UNIT

FIGURE 4·5



* Only one of these NOR gates are required at a time.

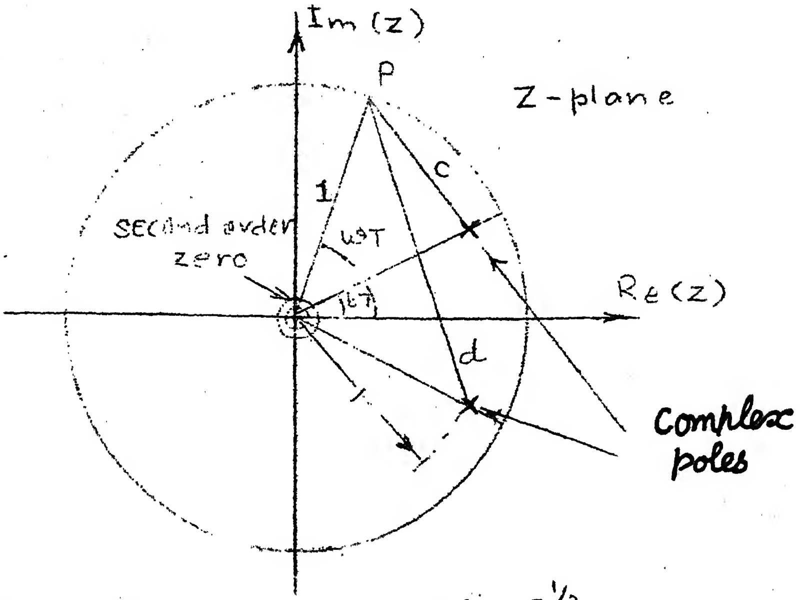
FIGURE 4·6 : DECODING SCHEME FOR PROGRAM COUNTER



$$Y = X - 2r\cos bT\,(X - YZ^{-1}) + r^2\,(X - YZ^{-2})$$

2-POLE DIGITAL FILTER

FIGURE A.1



$$c = [1 + r^2 - 2r\cos(\omega - b)T]^{1/2}$$

$$d = [1 + r^2 - 2r\cos(\omega + b)T]^{1/2}$$

Z-PLANE POLE ZERO DIAGRAM FOR DIGITAL FORMANT

FIGURE A.2